CLAIMS

What is claimed is:

1. A method of providing a protected execution environment on a computer comprising:

intercepting an input/output request for a file from an application;

determining if the application is authorized to modify the protected execution environment;

creating a redirected input/output request to an alternate environment when the application is not authorized to modify the protected execution environment and the file is within the protected execution environment; and

submitting the redirected input/output request to a file system manager.

2. The method of claim 1 further comprising:

allowing the redirected input/output request to continue when it is intercepted.

3. The method of claim 1 further comprising:

creating the protected execution environment.

4. The method of claim 1 wherein the protected execution environment comprises a directory for each of the applications that is authorized to modify the protected execution environment.

5. The method of claim 1 further comprising:

categorizing each application installed on the computer as authorized or not authorized to modify the protected execution environment.

6. The method of claim 1 wherein the alternate environment comprises a directory associated with an application that is not authorized to modify the protected execution environment.

7. The method of claim 1 wherein the redirected input/output request specifies a directory in the alternate environment that corresponds to a directory in the protected execution environment specified in the input/output request.

8. The method of claim 1, wherein a parent-child relationship is maintained between an application that invokes another application.

9. The method of claim 1, wherein determining if the application is authorized to modify the protected execution environment comprises:

designating the application as not authorized to modify the protected execution environment if the application was invoked by another application that is not authorized to modify the protected execution environment.

10. The method of claim 1, further comprising:

creating a null entry in a mirror directory structure for an executable for each application authorized to modify the protected execution environment,

wherein determining if the application is authorized to modify the protected execution environment comprises:

querying the existence of the executable for the application in the mirror directory structure.

11. The method of claim 10, further comprising:

maintaining an association between an executing application and a directory path for the executable for the executing application,

wherein querying for the existence of the executable in the mirror data structure comprises:

specifying the directory path for the executable associated with the executing application.

12. A method for operating a computer system with a protected execution environment comprising:

executing a configuration utility to categorize a plurality of applications installed on the computer system as authorized or not authorized to modify the protected execution environment;

defining the protected execution based on the authorized applications; and

installing a protected execution agent in a file system to intercept input/output requests submitted by the applications, wherein the protected execution agent directs an input/output request to an alternate environment if the application that submitted the request is not authorized and the request is directed to the protected execution environment.

13. The method of claim 12 wherein the configuration utility defines the protected execution environment when categorizing the plurality of applications.

14. The method of claim 12 wherein the alternate environment is defined based on at least one application that is not authorized.

15. The method of claim 12, wherein the alternate environment is defined by the configuration utility when categorizing the plurality of applications.

16. The method of claim 12, wherein the configuration utility further creates a null entry in a mirror directory structure for an executable for each authorized application and the protected execution agent further queries the existence of the executable for an executing application in the mirror directory structure to determine if the application is authorized.

17. The method of claim 16, wherein the protected execution agent further maintains an association between the executing application and a directory path for the executable for the executing application.

18. The method of claim 12, wherein the protected execution agent designates a second application as not authorized if it was invoked by a first application that is not authorized.

19. The method of claim 18, wherein the protected execution agent maintains a parent-child relationship between the first and second applications.

20. The method of claim 12, wherein the protected execution agent is installed in a hook chain in a file system manager to intercept the input/output requests before the requests are processed by any other agent installed in the hook chain.

21. The method of claim 12, wherein the configuration utility is executed prior to providing the computer system to a user and the protected execution agent is installed each time the computer system is booted.

22. The method of claim 12, further comprising:

saving a copy of the protected execution environment; and

recovering from a failure of the computer system by replacing the protected execution environment with the copy.

23. The method of claim 22, wherein the copy is saved on the computer system in a secure location.

24. The method of claim 22, wherein the copy is saved on a remote computer server and downloaded to the computer system.

25. A method of determining a category for an application on a computer comprising:

categorizing the application as a first type;

creating a directory in a second directory structure for the application when it is a first type, wherein the second directory structure mirrors a first directory structure that contains an executable for the application;

creating a null entry for the executable for the application in the directory in the second directory structure when the application is the first type;

querying the existence of the executable for the application in the second directory structure, wherein the application is determined to be the first type when the executable exists.

26. A computer-readable medium having stored thereon computer-executable instructions for performing a method comprising:

intercepting an input/output request for a file from an application;

determining if the application is authorized to modify the protected execution environment;

creating a redirected input/output request to an alternate environment when the application is not authorized to modify the protected execution environment and the file is within the protected execution environment; and

submitting the redirected input/output request to a file system manager.

27. The computer-readable medium of claim 26 having further computer-readable instructions comprising:

allowing the redirected input/output request to continue when it is intercepted.

28. The computer-readable medium of claim 26 having further computer-readable instructions comprising:

categorizing each application installed on the computer as authorized or not authorized to modify the protected execution environment.

29. The computer-readable medium of claim 26 having further computer-readable instructions comprising:

creating the protected execution environment from a directory for each of the applications that is authorized to modify the protected execution environment.

30. The computer-readable medium of claim 26 having further computer-readable instructions comprising:

creating the alternate environment from a directory associated with an application that is not authorized to modify the protected execution environment.

31. The computer-readable medium of claim 26 having further computer-readable instructions comprising:

storing a directory path specified in the input/output request in the redirected input/output request to direct the request to a corresponding directory path in the alternate environment.

32. The computer-readable medium of claim 26 having further computer-readable instructions comprising:

maintaining a parent-child data structure to track between relationships between applications that invoke other applications.

33. The computer-readable medium of claim 26 having further computer-readable instructions comprising:

designating the application as not authorized to modify the protected execution environment if the application was invoked by another application that is not authorized to modify the protected execution environment.

34. The computer-readable medium of claim 26 having further computer-readable instructions comprising:

creating a null entry in a mirror directory structure for an executable for each application authorized to modify the protected execution environment; and

querying the existence of the executable for the application in the mirror directory structure when determining if the application is authorized to modify the protected execution environment

35. The computer-readable medium of claim 34 having further computer-readable instructions comprising:

maintaining an association between an executing application and a directory path for the executable for the executing application; and

specifying the directory path for the executable associated with the executing application when querying for the existence of the executable in the mirror data structure.

36. A computer system comprising:

a processing unit;

a memory coupled to the processing unit through a system bus;

a computer-readable medium coupled to the processing through the system bus; and

a protected environment agent executing from the computer-readable medium, wherein the protected environment agent causes the processing unit to intercept input/output requests submitted by applications executing on the computer system and further causes the processing unit to redirect each input/output request to an alternate environment if the application that submitted the request is not authorized to modify a protected execution environment and the request is directed to the protected execution environment.

37. The computer system of claim 36 further comprising:

a configuration utility executing from the computer-readable medium, wherein the configuration utility causes the processing unit to categorize each application installed on the computer system as authorized or not authorized to modify the protected execution environment and further to cause the processing unit to define the protected execution environment to contain directories associated with the authorized applications.